https://github.com/Tuxae/Tuxae-Jupyter-Manager

Setup Jupyter environments for data scientists

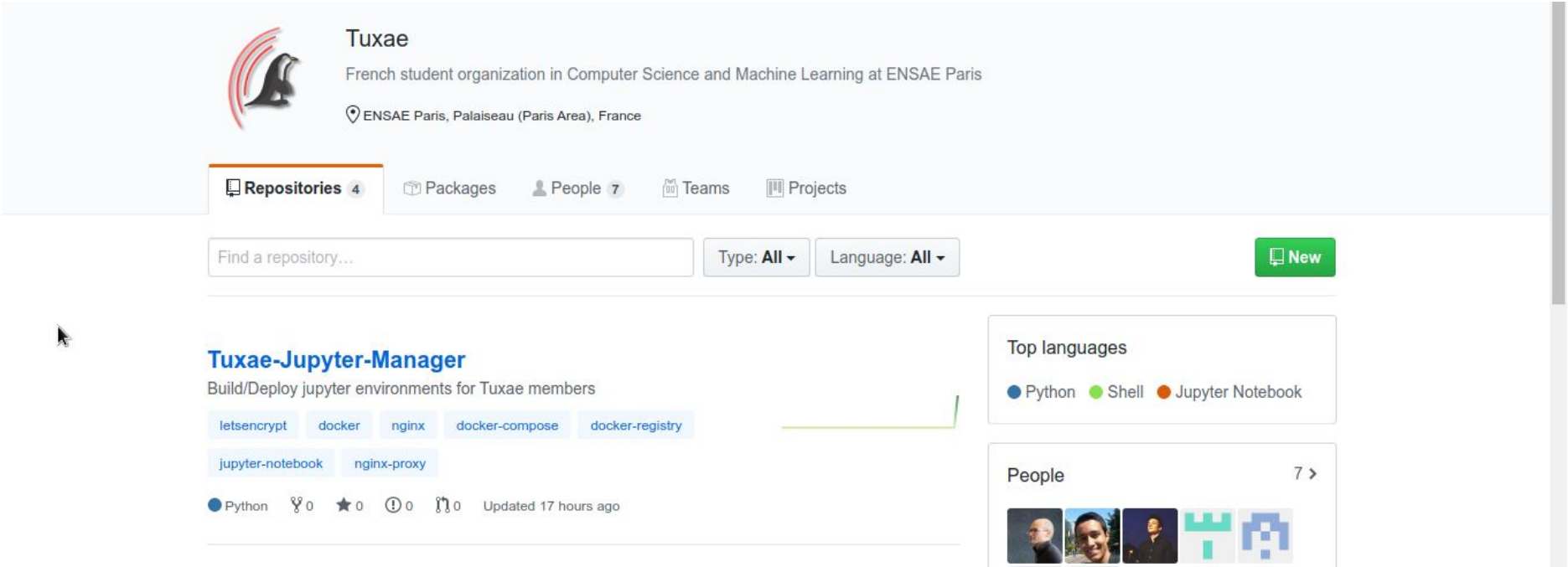
Technologies: Docker, Jupyter, Letsencrypt, Nginx proxy, Flask, UWSGI

ENSAE

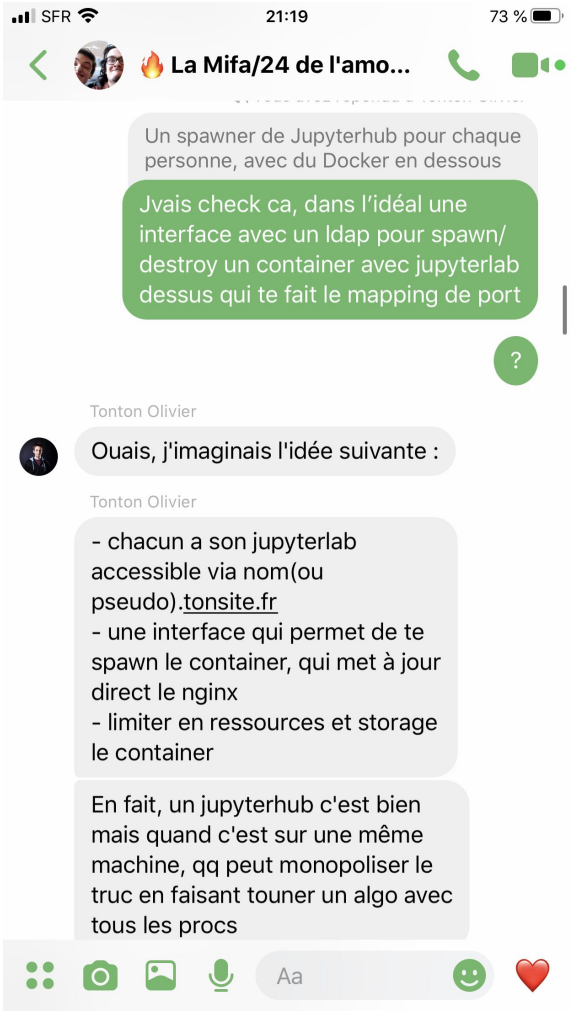
IP PARIS

Author: Aurélien Duboc <aurelien@duboc.xyz>

Github Tuxae



Last saturday



Finance application

jupyter nbviewer

JUPYTER FAQ

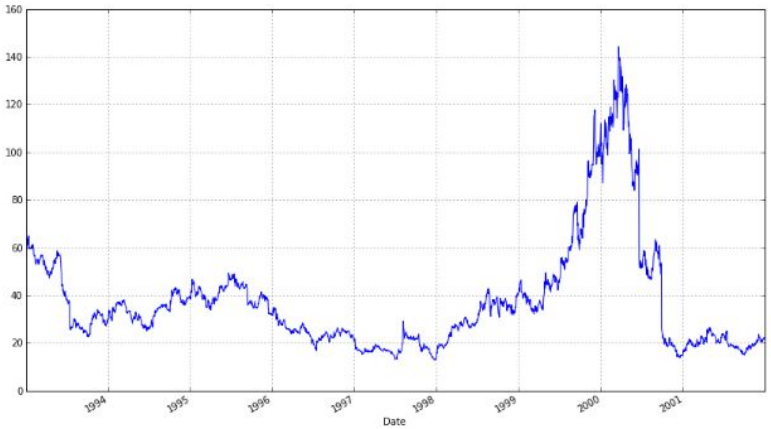
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
plt.figsize(14,8)

from zipline.algorithm import TradingAlgorithm
from zipline.transforms import MovingAverage, batch_transform
from zipline.utils.factory import load_from_yahoo
```

In [3]:
```
data = pd.load('talk_px.dat') #data = load_from_yahoo(stocks=['AAPL', 'PEP', 'KO']); data.save('talk_px.dat')
```

In [4]:
```
data['AAPL'].plot()
```

Out[4]: <matplotlib.axes.AxesSubplot at 0xa80784c>



In [5]:
```
# Simplest possible algorithm
class BuyApple(TradingAlgorithm): # inherit from TradingAlgorithm
    def handle_data(self, data): # overload handle_data() method
        self.order('AAPL', 1) # stock (='AAPL') to order and amount (=1 shares)
```

In [6]:
```
my_algo = BuyApple() # Instantiate our algorithm
results_buy_apple = my_algo.run(data) # Backtest algorithm on dataframe.
```

```
[2012-10-26 15:19] INFO: Performance: Simulated 2267.0 trading days out of 2268.
[2012-10-26 15:19] INFO: Performance: first open: 1993-01-04 14:30:00+00:00
```

In [7]:
```
print results_buy_apple
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2267 entries, 1993-01-04 21:00:00 to 2001-12-28 21:00:00
Data columns:
capital_used                2267  non-null values
cumulative_capital_used     2267  non-null values
ending_cash                 2267  non-null values
ending_value                2267  non-null values
max_capital_used            2267  non-null values
max_leverage                2267  non-null values
period_close                2267  non-null values
period_open                 2267  non-null values
pnl                         2267  non-null values
portfolio_value             2267  non-null values
positions                   2267  non-null values
returns                     2267  non-null values
starting_cash               2267  non-null values
starting_value              2267  non-null values
transactions                2267  non-null values
dtypes: float64(11), object(4)
```

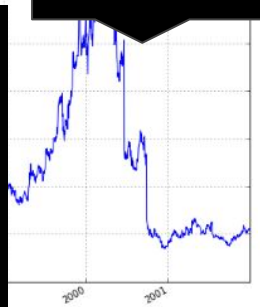
In [8]:
```
results_buy_apple.portfolio_value.plot()
```

Out[8]: <matplotlib.axes.AxesSubplot at 0xaf2c20c>

Finar

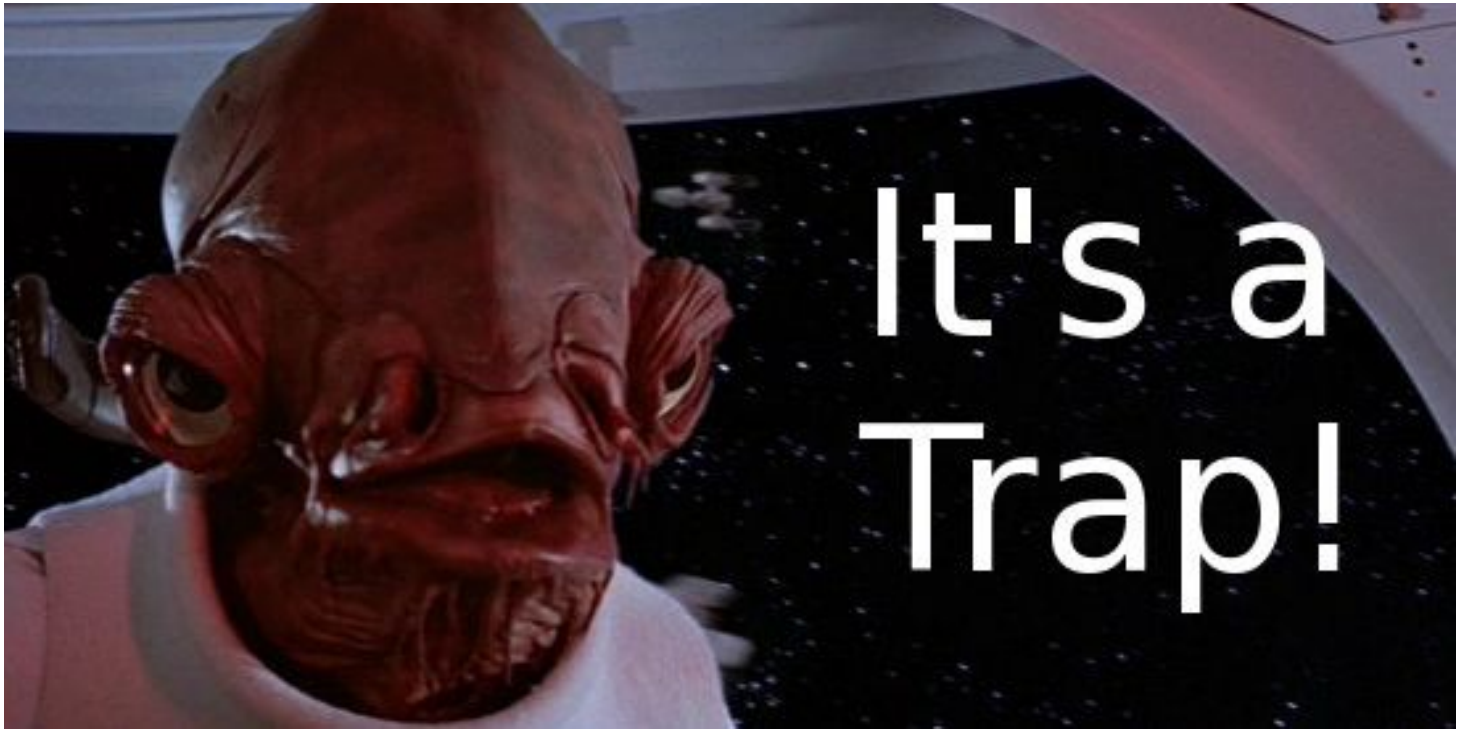
jupyter
nbviewer

TRO B1 LA THUNE



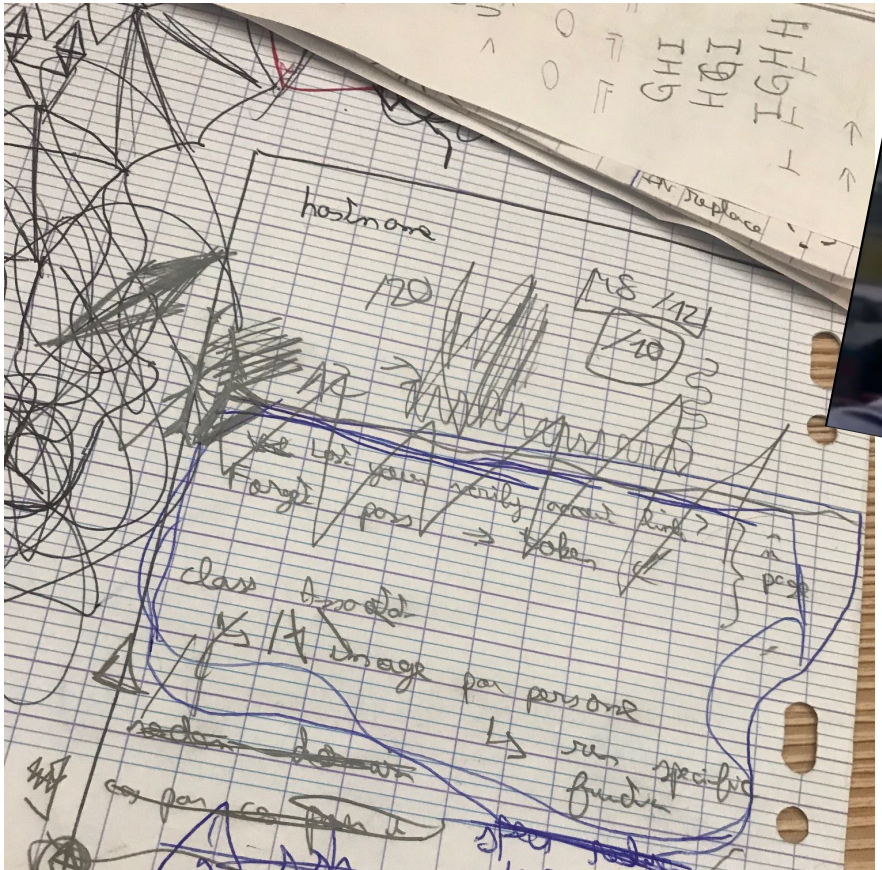
What do I need ??

Step1: a well-rested body for more energy

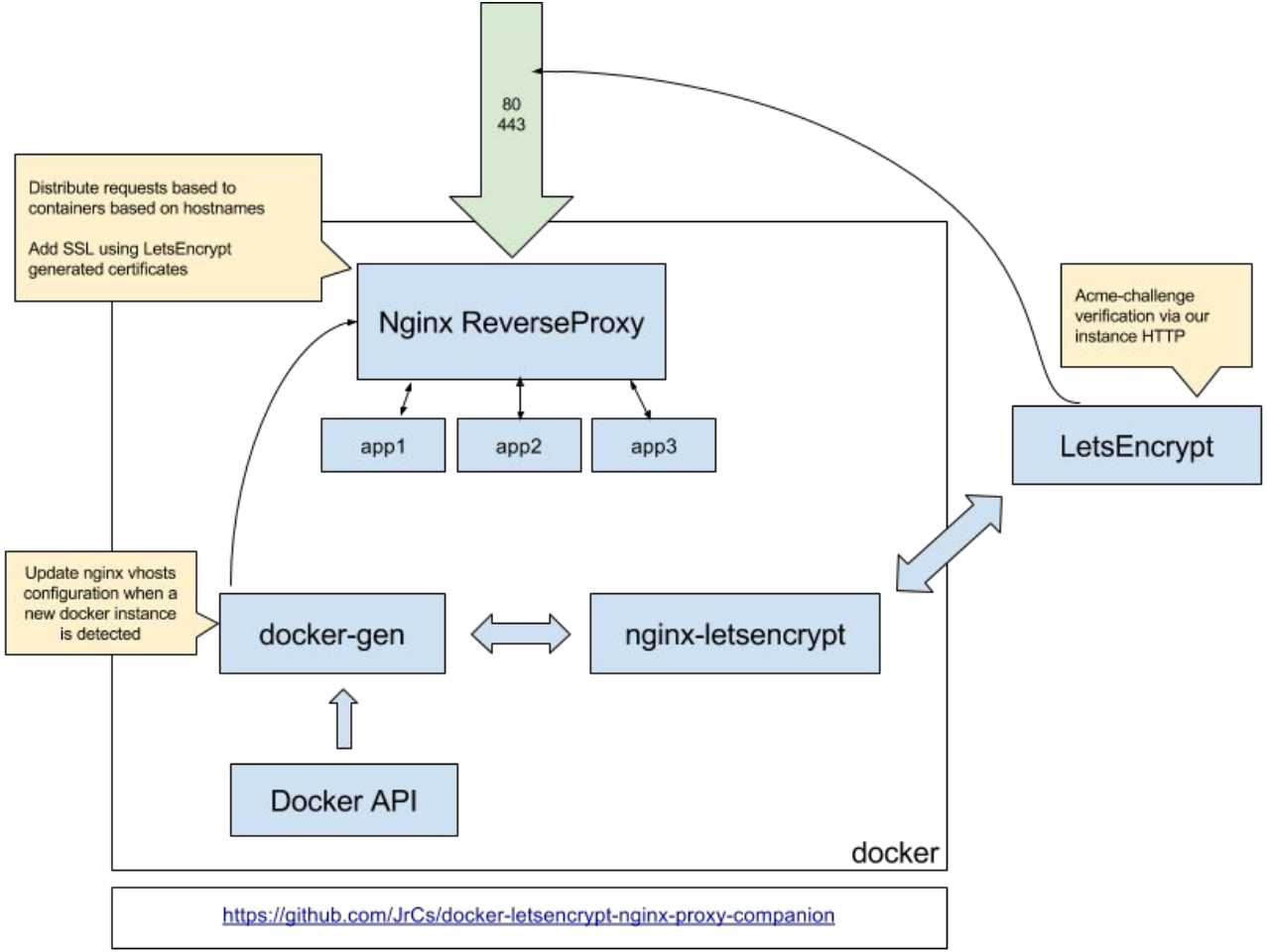


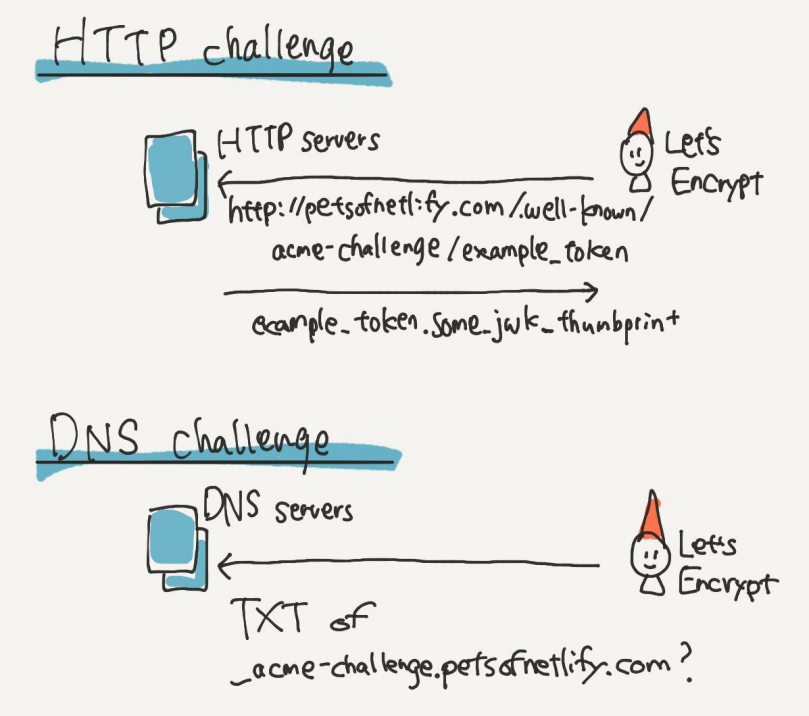
Step1: a we body for more energy

Step2: Take some organized notes



Let’s work !


80
443
Distribute requests based to containers based on hostnames
Add SSL using LetsEncrypt generated certificates
Nginx ReverseProxy
app1
app2
app3
Acme-challenge verification via our instance HTTP
LetsEncrypt
Update nginx vhosts configuration when a new docker instance is detected
docker-gen
nginx-letsencrypt
Docker API
docker
https://github.com/JrCs/docker-letsencrypt-nginx-proxy-companion

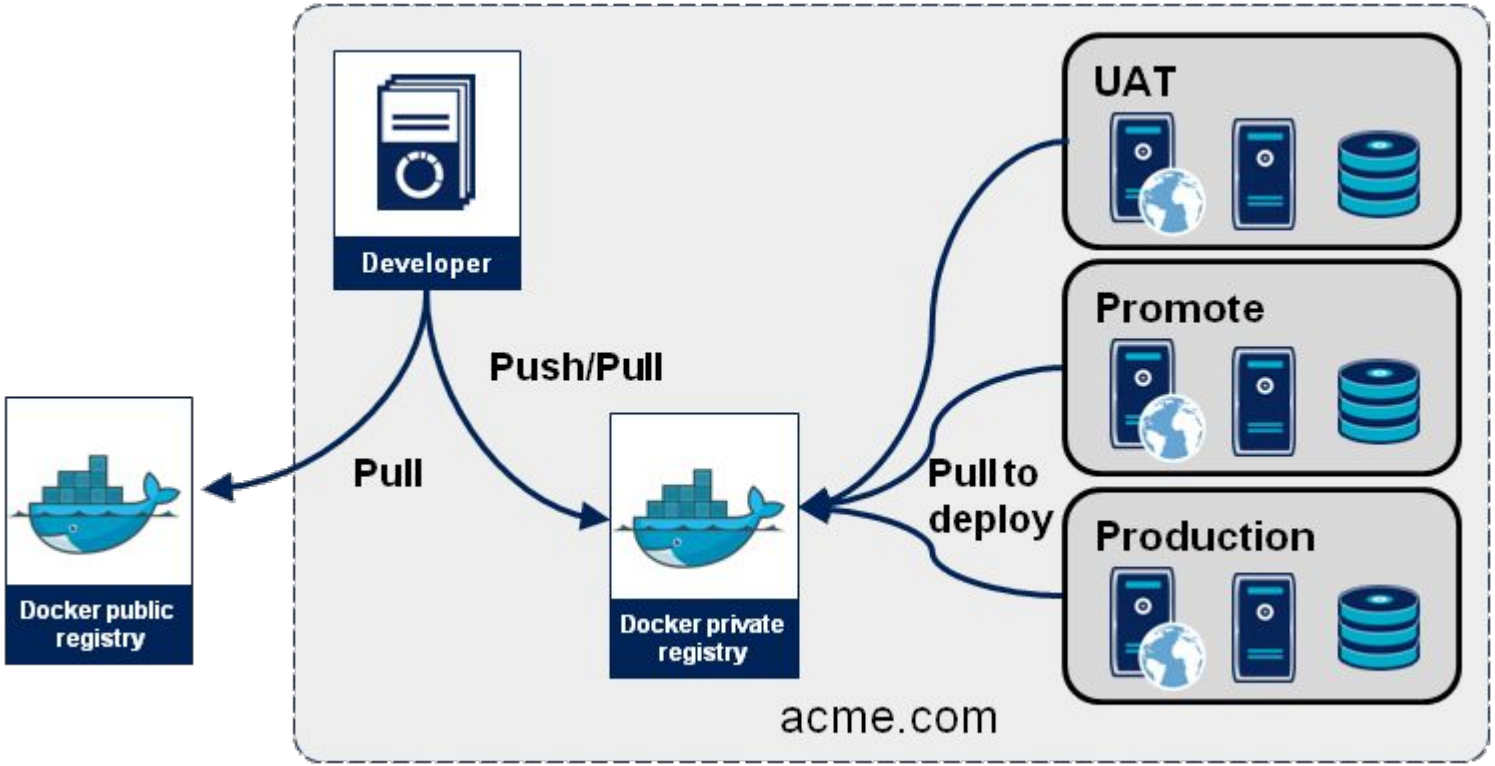


https://letsencrypt.org/fr/docs/challenge-types/

https://thecustomizewindows.com/2017/04/what-is-caa-dns-record-and-how-to-add/

DNS configuration

server.domain.tld.	0	A	157.159.191.54
*.domain.tld.	0	CNAME	server.domain.tld.
domain.tld.	0	CAA	0 issuewild "letsencrypt.org."


Developer
Push/Pull
Pull
Docker public registry
Docker private registry
Pull to deploy
UAT
Promote
Production
acme.com

jupyter/scipy-notebook

Source on GitHub | Dockerfile commit history | Docker Hub image tags

`jupyter/scipy-notebook` includes popular packages from the scientific Python ecosystem.

- Everything in `jupyter/minimal-notebook` and its ancestor images
- pandas, numexpr, matplotlib, scipy, seaborn, scikit-learn, scikit-image, sympy, cython, patsy, statsmodel, cloudpickle, dill, numba, bokeh, sqlalchemy, hdf5, vincent, beautifulsoup, protobuf, and xlrd packages
- ipywidgets for interactive visualizations in Python notebooks
- Facets for visualizing machine learning datasets

jupyter/tensorflow-notebook

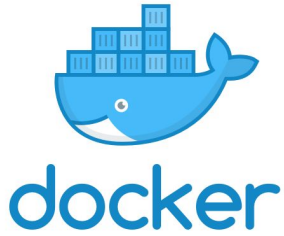
Source on GitHub | Dockerfile commit history | Docker Hub image tags

`jupyter/tensorflow-notebook` includes popular Python deep learning libraries.

- Everything in `jupyter/scipy-notebook` and its ancestor images
- tensorflow and keras machine learning libraries

jupyter/datascience-notebook

Source on GitHub | Dockerfile commit history | Docker Hub image tags



Docker registry

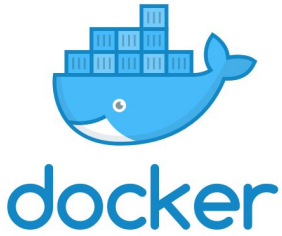
The web application provide an access to docker images from the local registry

Example to add image to the local registry:

```
docker pull jupyter/datascience-notebook:latest
docker tag jupyter/datascience-notebook:latest 127.0.0.1:5000/datascience-notebook
docker push 127.0.0.1:5000/datascience-notebook
```

You can check available images on http://127.0.0.1:5000/v2/_catalog

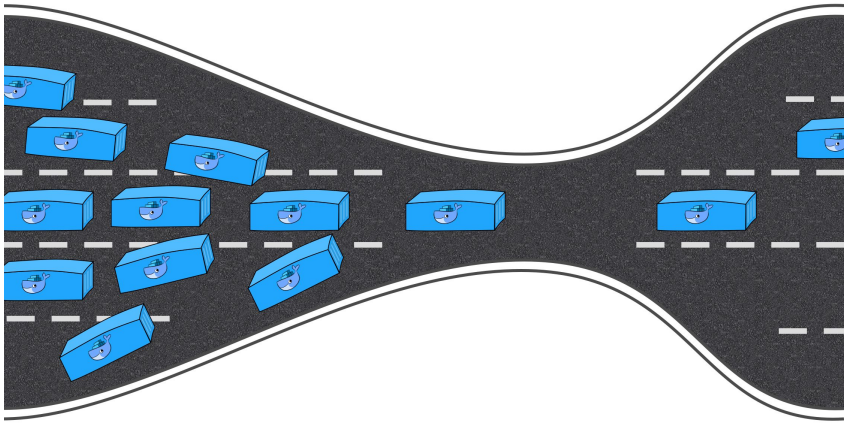
```
curl http://127.0.0.1:5000/v2/_catalog
{"repositories":["datascience-notebook"]}
```

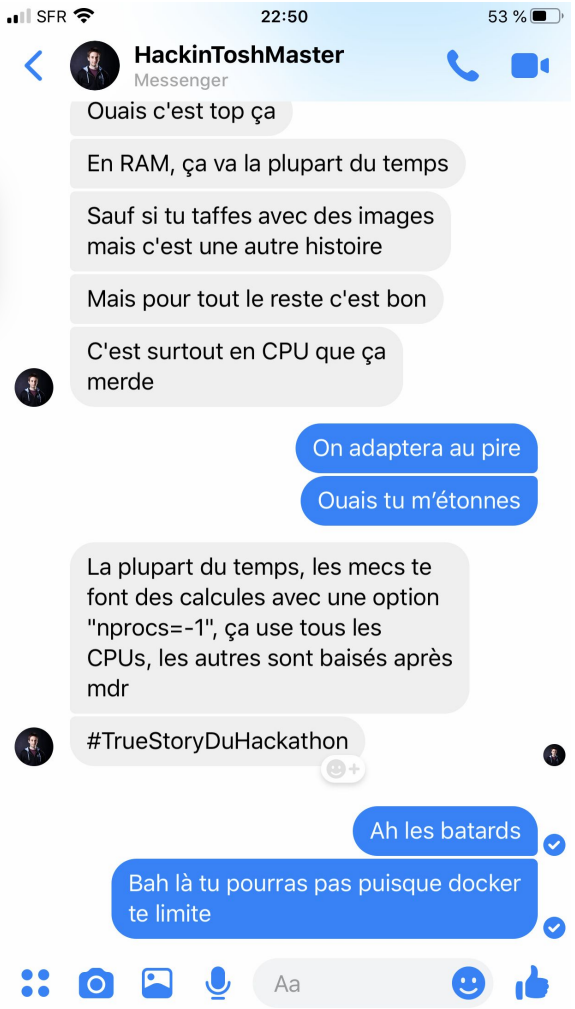


CPU/RAM

srainville — ubuntu@ubuntu16: ~ — ssh docker01 — 151×5

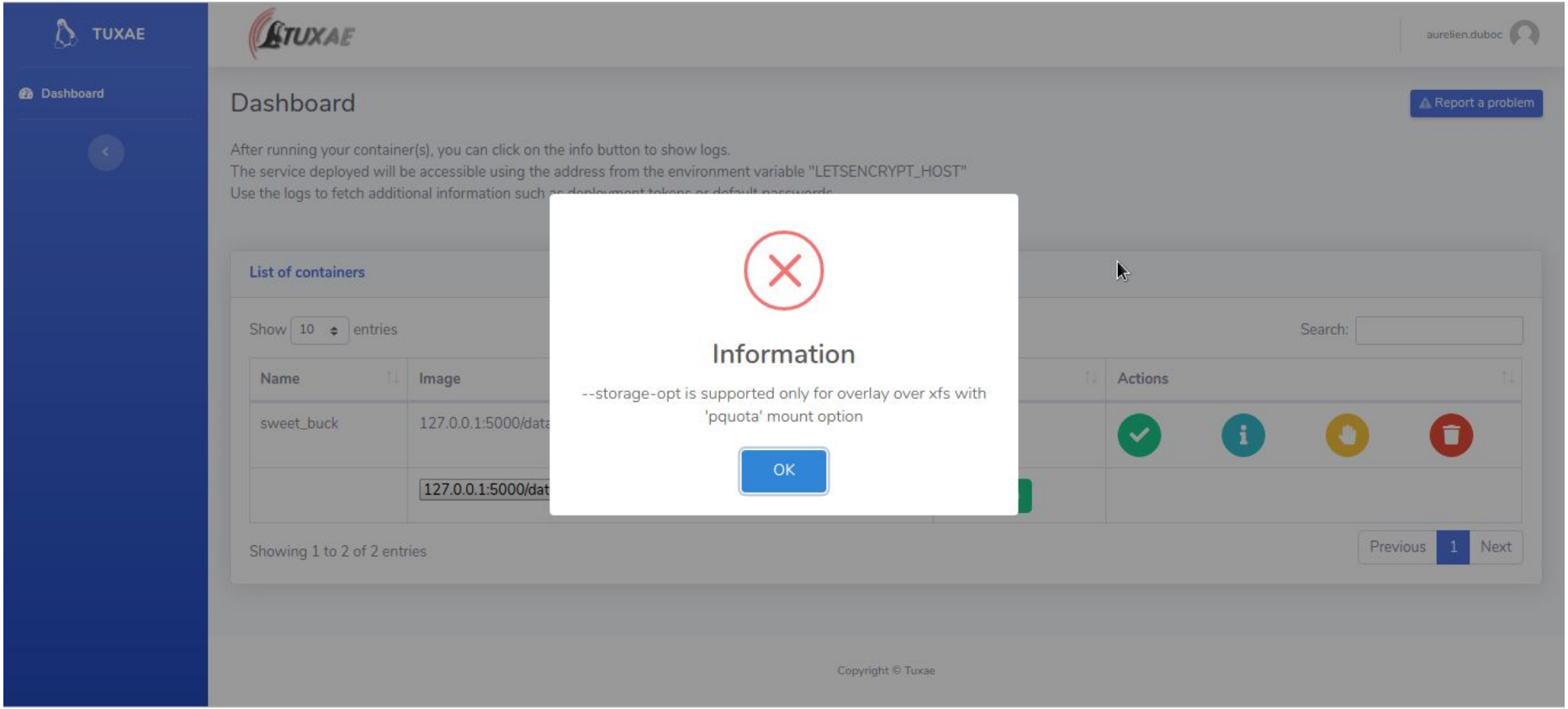
CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
1c7eb5235562	eloquent_morse	0.00%	2.352MiB / 256MiB	0.92%	648B / 0B	0B / 0B	2
08b3ba3b8d08	happy_poitras	0.00%	1.344MiB / 7.796GiB	0.02%	828B / 0B	0B / 0B	2





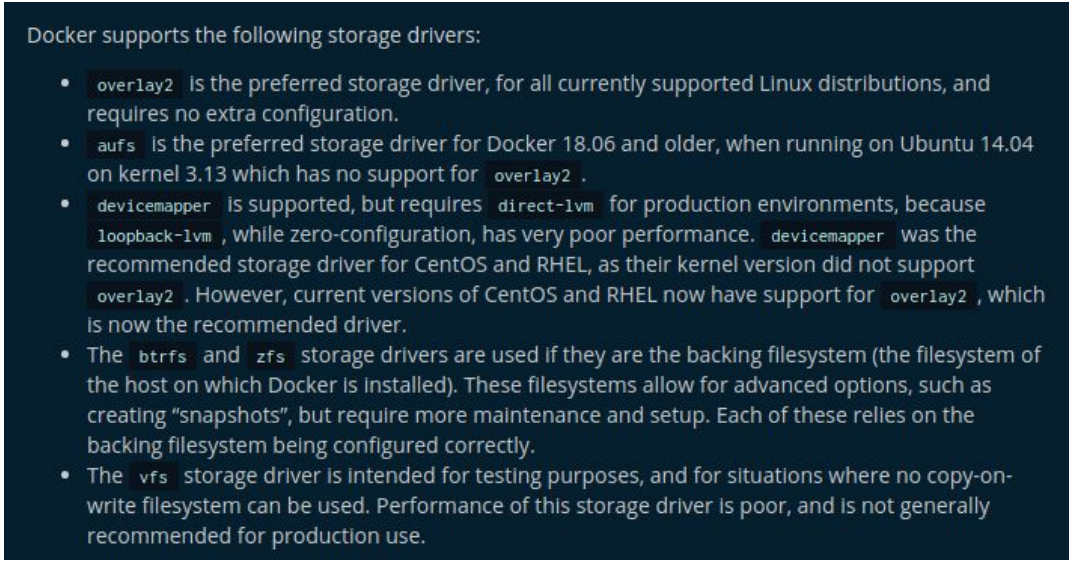
TUXAE

Storage limit issue



Fixing storage issue

1. In computing, OverlayFS is a file system for Linux. He has encountered a Union Mount for other file systems. It was merged into the Linux kernel in 2014, in kernel version 3.181.2.
2. XFS is a high-performance 64-bit log file system created by SGI for its IRIX operating system. In May 2000, SGI places XFS under the GPL license. XFS was ported to the Linux kernel in 2001. Since June 2014, XFS is supported by most Linux distributions, some of which are used as the default file system.



Docker supports the following storage drivers:

- `overlay2` is the preferred storage driver, for all currently supported Linux distributions, and requires no extra configuration.
- `aufs` is the preferred storage driver for Docker 18.06 and older, when running on Ubuntu 14.04 on kernel 3.13 which has no support for `overlay2`.
- `devicemapper` is supported, but requires `direct-lvm` for production environments, because `loopback-lvm`, while zero-configuration, has very poor performance. `devicemapper` was the recommended storage driver for CentOS and RHEL, as their kernel version did not support `overlay2`. However, current versions of CentOS and RHEL now have support for `overlay2`, which is now the recommended driver.
- The `btrfs` and `zfs` storage drivers are used if they are the backing filesystem (the filesystem of the host on which Docker is installed). These filesystems allow for advanced options, such as creating "snapshots", but require more maintenance and setup. Each of these relies on the backing filesystem being configured correctly.
- The `vfs` storage driver is intended for testing purposes, and for situations where no copy-on-write filesystem can be used. Performance of this storage driver is poor, and is not generally recommended for production use.

3. A disk quota is a limit set by a system administrator that restricts certain aspects of file system usage on modern operating systems. The function of setting quotas to disks is to allocate limited disk-space in a reasonable way.

https://wiki.archlinux.org/index.php/Disk_quota

Fixing storage issue

Storage quota per container - overlay2 backed by xfs

Open Source Projects | DockerEngine | docker

vishr Vishal Rana — Sep '17

Hi,

In my setup, docker daemon is running with overlay2 storage backed by xfs. Following this https://docs.docker.com/engine/reference/commandline/run/#set-storage-driver-options-per-container 248 document I found that I need to mount `/` with pqouta option to able storage quota per container, so I edited `/etc/fstab` to include pquota option like `/dev/md1 / xfs defaults,pquota 0 0`, but I am not able to run a container like below

```
root@n1:~# docker run -it --storage-opt size=120G fedora /bin/bash
docker: Error response from daemon: --storage-opt is supported only for overlay over xfs with
'pquota' mount option.
See 'docker run --help'.
```

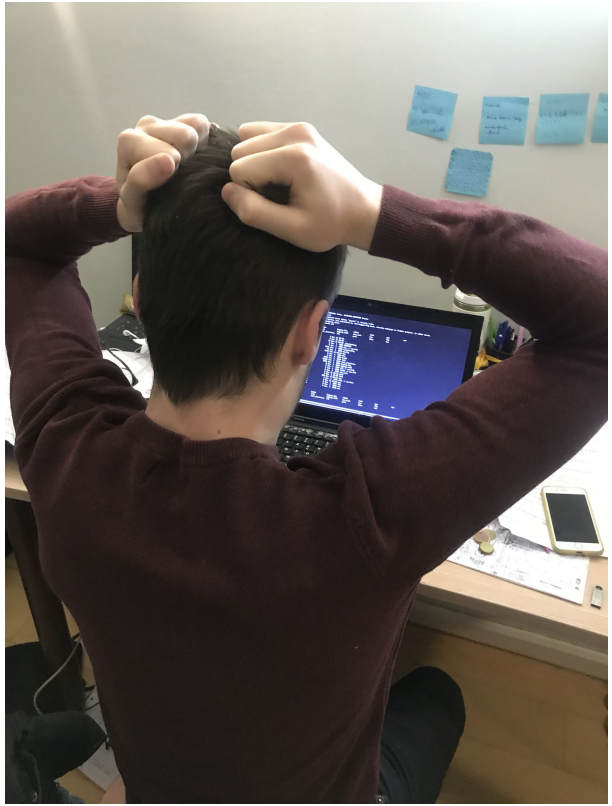
Do I need to follow any other steps?

Solved by vishr in post #2

I had to update /etc/default/grub with the following entry:
GRUB_CMDLINE_LINUX_DEFAULT="rootflags=uquota,pquota"

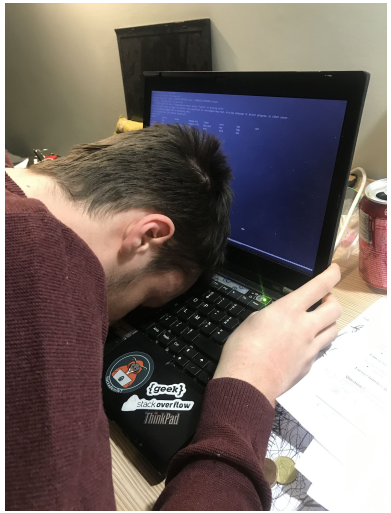
TUXAE

Fixing storage issue



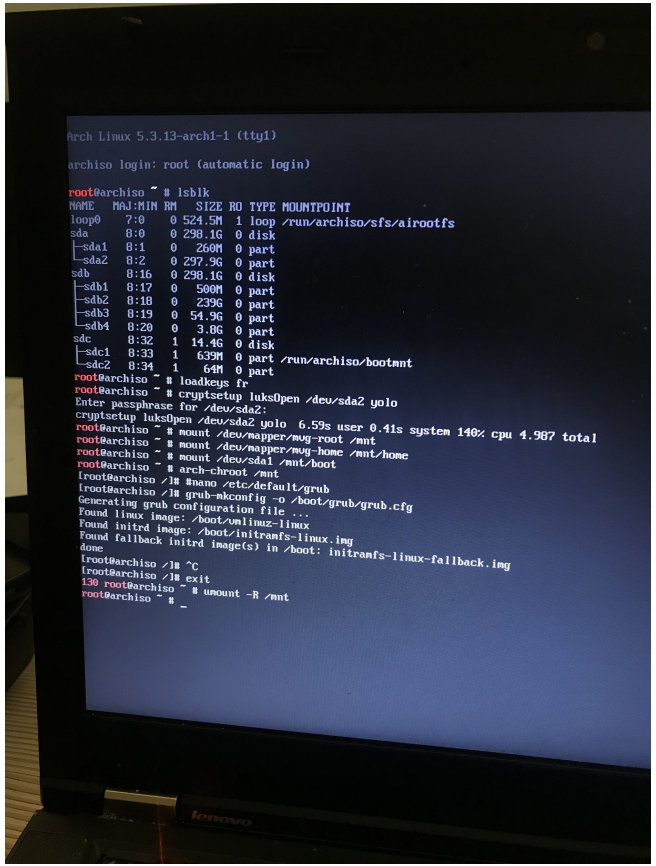
```
A password is required to access the cryptlvm volume:
Enter passphrase for /dev/sda2:
:: performing fsck on '/dev/mvg/root'
/dev/mapper/mvg-root: clean, 1633034/4587520 files, 15006968/18350080 blocks
:: mounting '/dev/mvg/root' on real root
[    9.477293] EXT4-fs (dm-2): Unrecognized mount option "uquota" or missing value
mount: /new_root: wrong fs type, bad option, bad superblock on /dev/mapper/mvg-root, missing codepage or helper program, or other error.
You are now being dropped into an emergency shell.
sh: can't access tty; job control turned off
[rootfs ]# ls
VERSION         config          hooks           keymap.bin      lib64           root            sys             var
bin             dev             init            keymap.utf8     new_root        run             tmp
buildconfig     etc             init_functions  lib             proc            sbin            usr
```

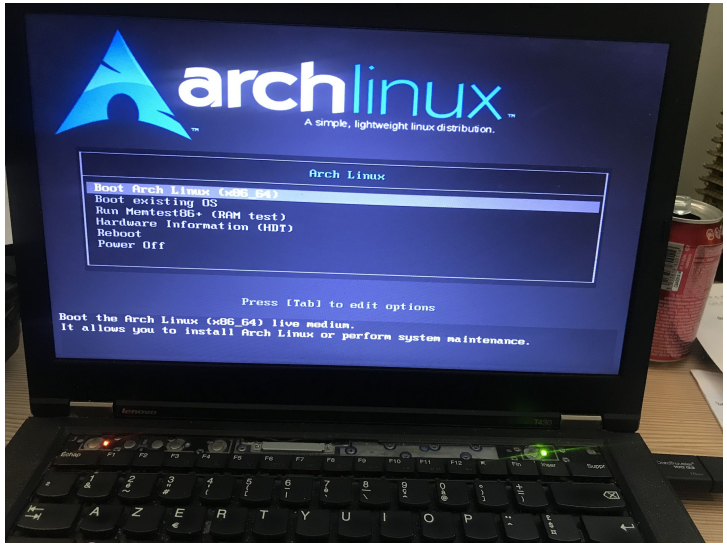
CANNOT BOOT !!



TUXAE

Fixing storage issue





Trying to fix this mess from live OS

Then reboot and pray...

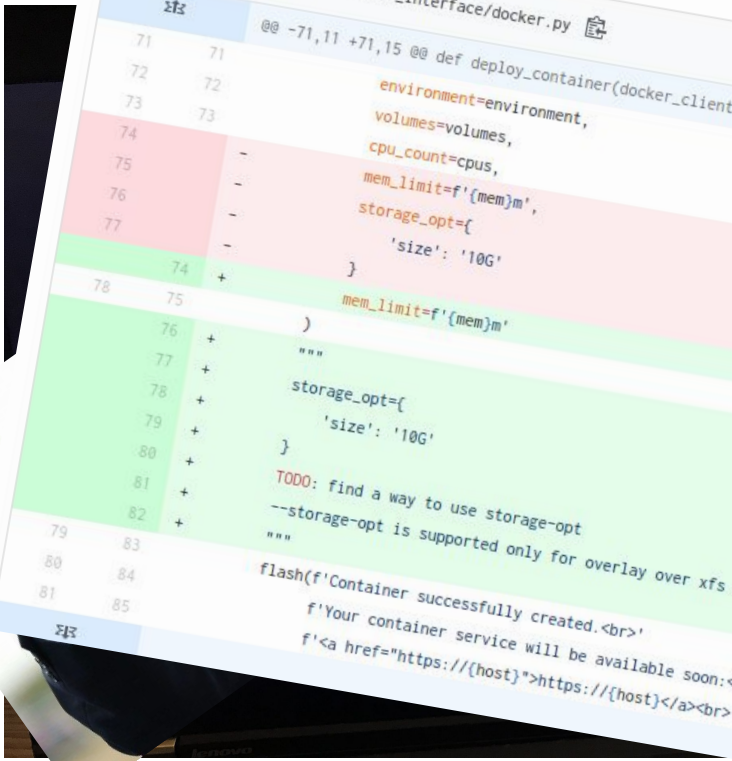
TUXAE

Fixing storage issue

System fixed but no solution...

TUXAE

Fixi



zteeed committed 2 minutes ago

Showing **1 changed file** with **8 additions** and **4 deletions**.

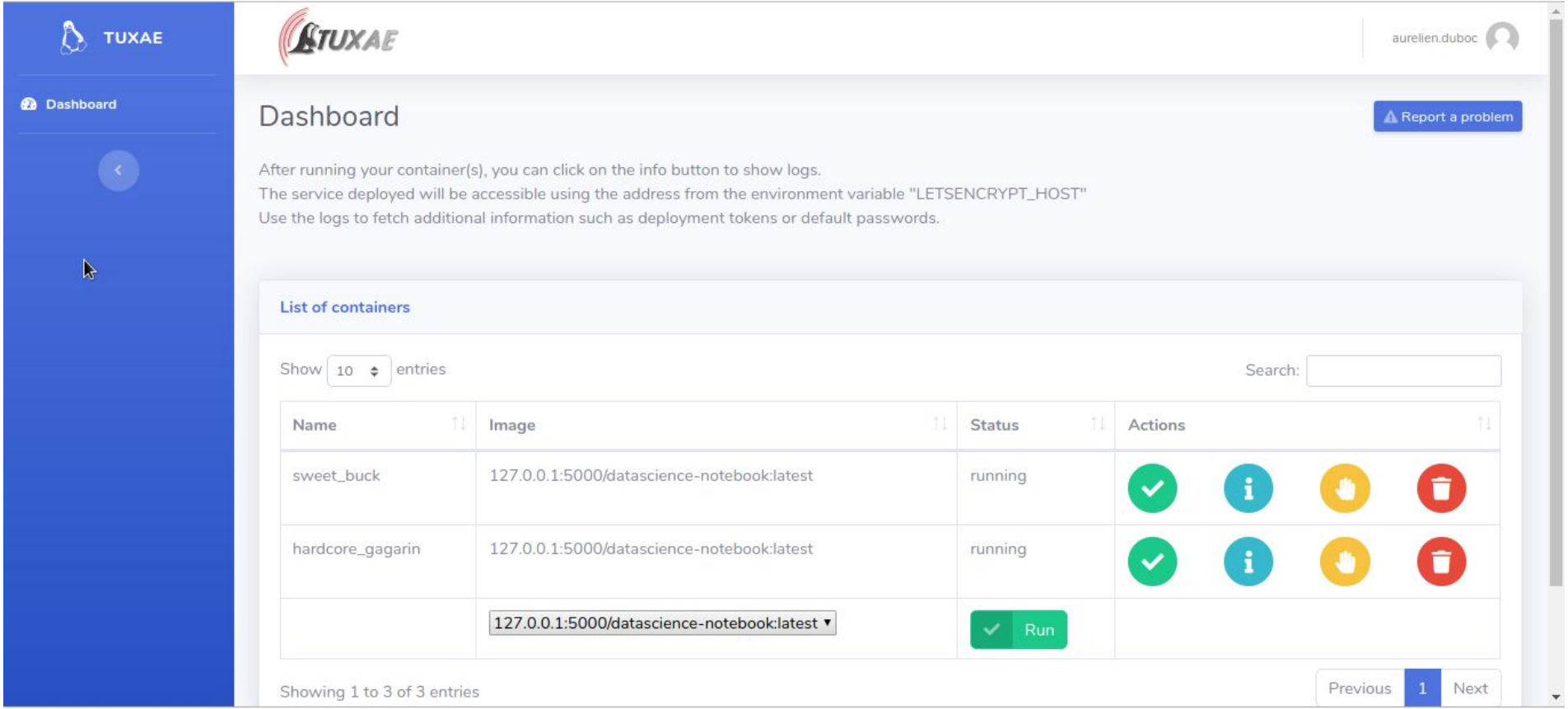
1 parent 1cd0898 commit fddb531dc28ec9038e50e6378182acd6b11eb405

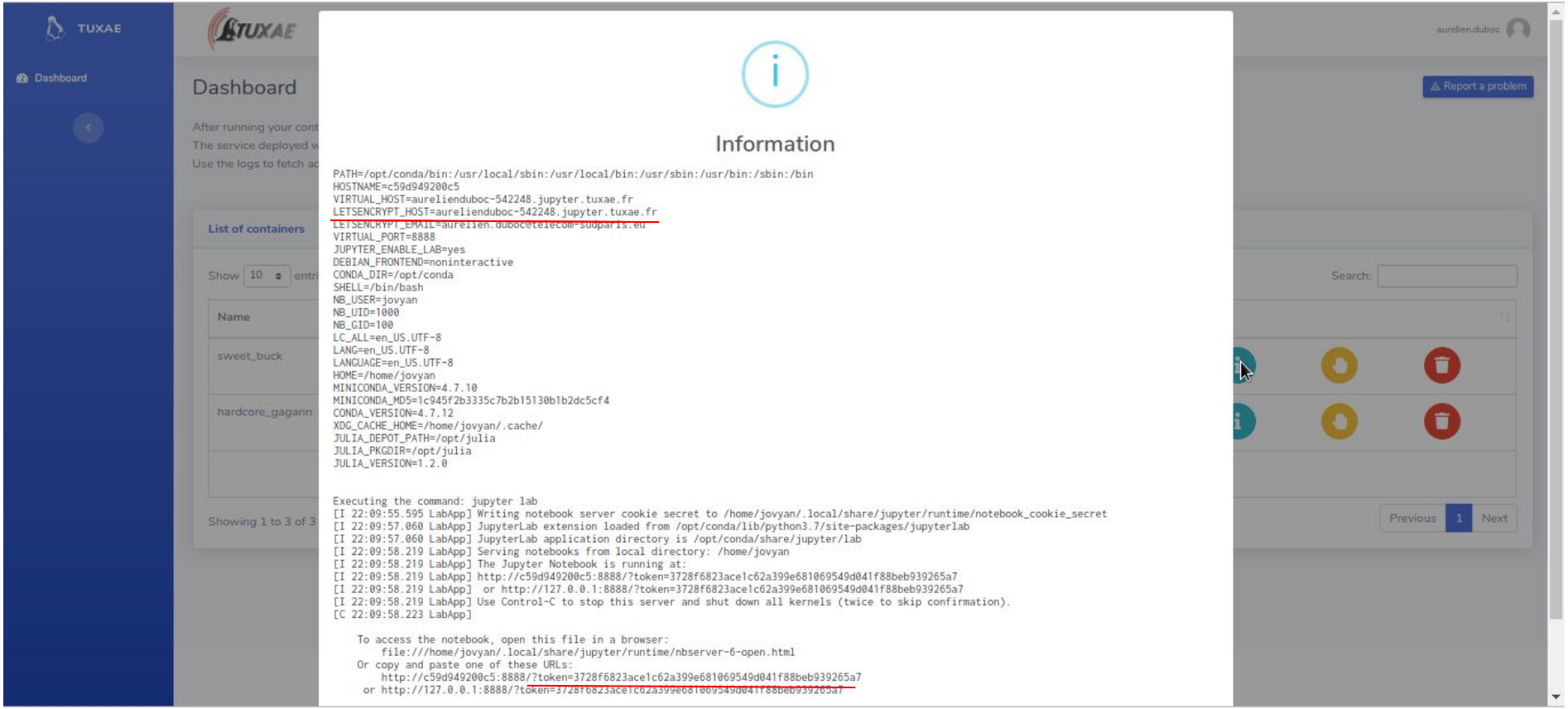
12 app/src/docker_interface/docker.py

Unified Split

```
@@ -71,11 +71,15 @@ def deploy_container(docker_client: docker.client.DockerClient, image: str, curr
            environment=environment,
            volumes=volumes,
-           cpu_count=cpus,
-           mem_limit=f'{mem}m',
-           storage_opt={
-               'size': '10G'
-           }
+           mem_limit=f'{mem}m'
        )
+       """
+       storage_opt={
+           'size': '10G'
+       }
+       TODO: find a way to use storage-opt
+       --storage-opt is supported only for overlay over xfs with 'pquota' mount option
+       """
        flash(f'Container successfully created.<br>'
              f'Your container service will be available soon:<br>'
              f'<a href="https://{host}">https://{host}</a><br>'
```

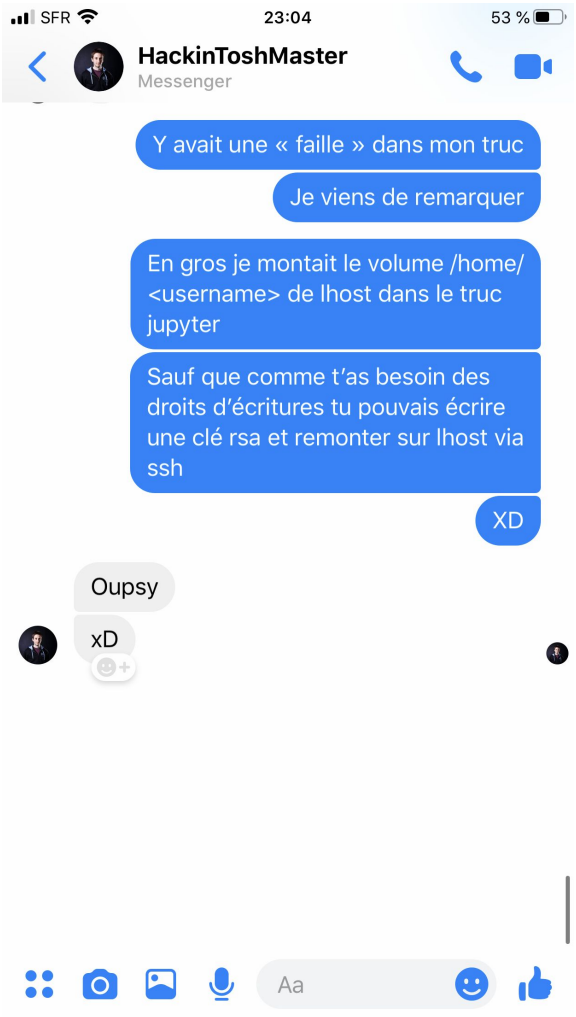
olution...


TUXAE
Dashboard
TUXAE
aurelien.duboc
Dashboard
Report a problem
After running your container(s), you can click on the info button to show logs.
The service deployed will be accessible using the address from the environment variable "LETSENCRYPT_HOST"
Use the logs to fetch additional information such as deployment tokens or default passwords.
List of containers
Show 10 entries
Search:
Name
Image
Status
Actions
sweet_buck
127.0.0.1:5000/datascience-notebook:latest
running
hardcore_gagarin
127.0.0.1:5000/datascience-notebook:latest
running
127.0.0.1:5000/datascience-notebook:latest
Run
Showing 1 to 3 of 3 entries
Previous
1
Next


TUXAE
Dashboard
Dashboard
aurelien.duboc
Report a problem
List of containers
Show 10 entri
Search:
Name
sweet_buck
hardcore_gagarin
Showing 1 to 3 of 3
Previous
1
Next
Information
PATH=/opt/conda/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=c59d949200c5
VIRTUAL_HOST=aureliendubоc-542248.jupyter.tuxae.fr
LETSENCRYPT_HOST=aureliendubоc-542248.jupyter.tuxae.fr
LETSENCRYPT_EMAIL=aurelien.duboc@telecom-sudparis.eu
VIRTUAL_PORT=8888
JUPYTER_ENABLE_LAB=yes
DEBIAN_FRONTEND=noninteractive
CONDA_DIR=/opt/conda
SHELL=/bin/bash
NB_USER=jovyan
NB_UID=1000
NB_GID=100
LC_ALL=en_US.UTF-8
LANG=en_US.UTF-8
LANGUAGE=en_US.UTF-8
HOME=/home/jovyan
MINICONDA_VERSION=4.7.10
MINICONDA_MD5=1c945f2b3335c7b2b15130b1b2dc5cf4
CONDA_VERSION=4.7.12
XDG_CACHE_HOME=/home/jovyan/.cache/
JULIA_DEPOT_PATH=/opt/julia
JULIA_PKGDIR=/opt/julia
JULIA_VERSION=1.2.0
Executing the command: jupyter lab
[I 22:09:55.595 LabApp] Writing notebook server cookie secret to /home/jovyan/.local/share/jupyter/runtime/notebook_cookie_secret
[I 22:09:57.060 LabApp] JupyterLab extension loaded from /opt/conda/lib/python3.7/site-packages/jupyterlab
[I 22:09:57.060 LabApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 22:09:58.219 LabApp] Serving notebooks from local directory: /home/jovyan
[I 22:09:58.219 LabApp] The Jupyter Notebook is running at:
[I 22:09:58.219 LabApp] http://c59d949200c5:8888/?token=3728f6823ace1c62a399e681069549d041f88beb939265a7
[I 22:09:58.219 LabApp] or http://127.0.0.1:8888/?token=3728f6823ace1c62a399e681069549d041f88beb939265a7
[I 22:09:58.219 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 22:09:58.223 LabApp]
To access the notebook, open this file in a browser:
file:///home/jovyan/.local/share/jupyter/runtime/nbserver-6-open.html
Or copy and paste one of these URLs:
http://c59d949200c5:8888/?token=3728f6823ace1c62a399e681069549d041f88beb939265a7
or http://127.0.0.1:8888/?token=3728f6823ace1c62a399e681069549d041f88beb939265a7

Apercu du Jupyter

Security issue



Security issue

Prevent user to access /home from host (used by sshd service)

master

zteeed committed 4 hours ago

1 parent 95656ae commit 9895c9d7f0a8a295caf80e8a48b5c1aad5e307da

Browse files

Unified Split

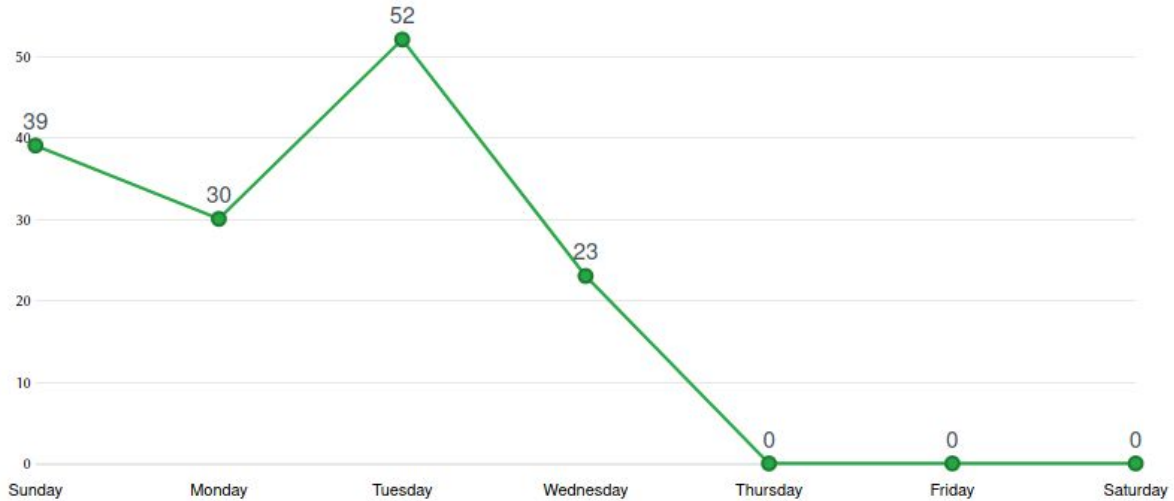
Showing **1 changed file** with **1 addition** and **1 deletion**.

2 app/src/docker_interface/docker.py

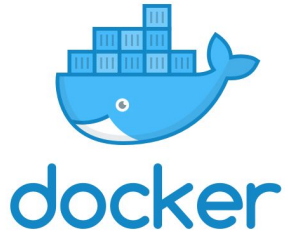
```
@@ -52,7 +52,7 @@ def deploy_container(docker_client: docker.client.DockerClient, image: str, curr
            f'JUPYTER_ENABLE_LAB=yes'
        ]
        volumes = {
-           f'/home/{username}':
+           f'/mnt/{username}':
                {
                    'bind': '/home/jovyan/work',
                    'mode': 'rw'
```

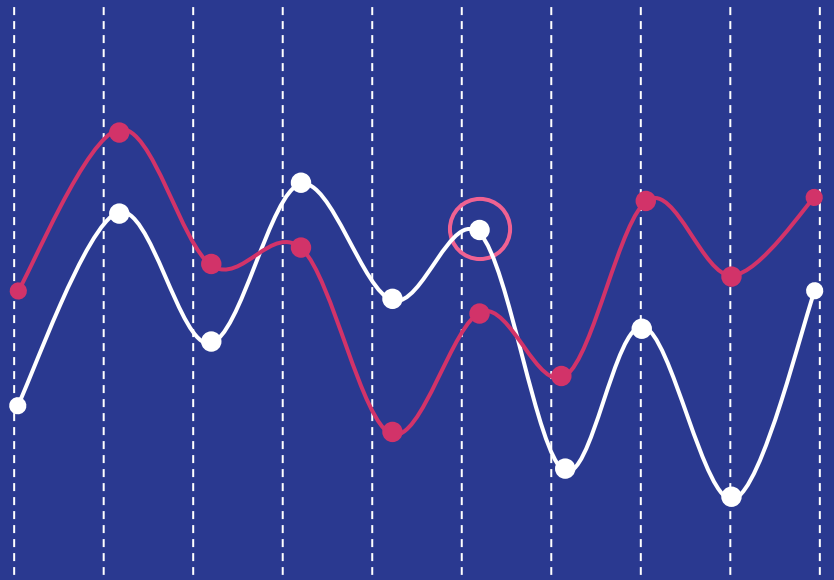
Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

- Pulse
- Contributors
- Community
- Traffic
- **Commits**
- Code frequency
- Dependency graph
- Network
- Forks



Thank You !





TUXAE